

# Design of Ultra-Low Power Consumption Approximate 4–2 Compressors based on the Compensation Characteristic

<sup>1</sup>P. Sahithi, <sup>2</sup>D. Shirisha, <sup>\*3</sup>Madipalli Sumalatha

<sup>1,2,3</sup>Department of ECE,

<sup>1,2,3</sup>Siddhartha Institute of Technology & Sciences, Narapally, Ghatkesar, Medchal-Malkajgiri, Telangana, India

<sup>\*3</sup>[sitsmtechms@gmail.com](mailto:sitsmtechms@gmail.com)

**Abstract:** Applications in digital signal processing that are inherently tolerant of inaccurate computing results are the ones that first use approximate computing. These circuits' electrical performance is enhanced by utilising the approximate arithmetic blocks. Some of the most basic building elements of computer mathematics are multipliers. In addition, the parallel multipliers frequently use the 4-2 compressors to speed up the partial product compression process. Three novel approximate 4-2 compressors used in 8-bit multipliers are shown in this letter. The approximation multiplier's error performance is enhanced when an error-correcting module (ECM) is paired with the recommended 4-2 compressors. An improved level of energy efficiency is brought about by this short by halving the number of outputs from the projected 4-2 compressor. Simulated findings show that the suggested approximation compressors UCAC1, UCAC2, and UCAC3 reduce latency by 24.76%, 51.43%, and 66.67%, power by 71.76%, 83.06%, and 93.28%, and area by 54.02%, 79.32%, and 93.10%, respectively, as compared to the exact 4-2 compressors. And using these suggested compressors in 8-bit multipliers results in an average power consumption reduction of 49.29%.

**Index Terms - Ultra-Low Power, Compensation Characteristic, 4–2 Compressors.**

## I. INTRODUCTION

The need for smart mobile devices (SMDs) with low power consumption is becoming more pressing as the IoT and edge computing continue to expand at a rapid pace, mainly because of their limited battery life [1]. New developments in low-power architecture, approximate computing, are having a significant impact in areas including robot vision, video surveillance, and image processing [2]. Applications that can handle mistakes and imprecision can nevertheless benefit from approximate computing, which loosens up on full precision while still producing valuable results [3]. The result is a hardware platform that is easier to run and performs better. Aside from being very power efficient, approximate-computation hardware offers superior performance when compared to precise logic circuits [4,5].

The multiplier is an essential mathematical unit in central processing units (CPUs) and has numerous uses, such as in filtering and convolutional neural networks. A lot of work has gone into finding ways to approximate computation using multipliers as of late [6]. Most of it has focused on one of three primary areas: partial product (PP) creation, PP trees, or compressors.

Nevertheless, nearly all positive errors are produced by the 4-2 compressors that were previously stated. We develop bias-free compressors that produce zero-sum positive and negative errors. We use the error reduction produced by the compressors to decrease the time latency of the key path in order to achieve high performance. To eliminate the temporal latency of the accuracy-part circuit, which affects the overall multiplier, we remove all but the most significant bits (MSBs) from n-1 to n-3. Both UBAM-M1 and UBAM-M2 use biased compressors as their approximation approaches.

The following are the contributions made by this paper.

- A compressor and the proposed approximation full adder can be used to build an approximate 6-2 compressor by balancing the signs of the errors in each compressor.
- The proposed unbiased approximation for a 4-2 compressor increases the likelihood that other 4-2 compressors in the same multiplier column would also create opposite sign errors, thereby lowering its error, and it also provides balanced positive and negative sign mistakes.
- By utilising the suggested approximate compressors, the UBAM-M1 and UBAM-M2 unbiased approximation multipliers are built. The performance of UBAM-M1 is enhanced compared to its predecessors due to the fact that approximation multipliers are now more precise. Approximate multipliers consume less power with UBAM-M2. There is a 39% improvement in the power-delay product (PDP) and a 46% improvement in the energy-delay product (EDP) compared to earlier designs. Minimising space by 26%, power consumption by 28%, and delay by 22% are all benefits of this.

## II. LITERATURE REVIEW

Regarding the advancement of partial products, it was noted that a  $2 \times 2$  multiplier module, which was associated with a Karnaugh map, had its output changed by an under designed multiplier (UDM). Additionally, to approximate the partial product, the  $2 \times 2$  multiplier was used [8]. Wide multipliers with N bits of high width are now also within the scope of this method's application. Carry prediction during PP buildup was the main focus of the study's UDM. Exploration of architectural space using an approximate multiplier with a maximum error value half of UDM was carried out. In contrast, the greatest mistake was more common, leading to a poorer outcome when delay was taken into account.

In order to reduce chip space and power consumption, PP tree approaches shorten the least relevant bits in partial products and remove the adders that go along with them. Because of this, truncation issues become more apparent to the audience. Following the aforementioned steps of rounding the multiplier result to  $n$  bits and truncating the least significant bits, a correction constant is used to replace the truncated bits. The result is then computed using an estimated value of the mistake [9].

Finally, make use of suitable correction functions, variable correction, and linear compensation to reduce the truncation error. PP trees are used in many different architectures, such as the AWTM, ETM, SSM, and SDLC (significance-driven logic compression) [10].

The three most common methods for building approximate arithmetic circuits are logic simplification, truncation, and voltage over scaling (VOS). Prior work on approximation addresses was substantial, yielding substantial gains in power and area but also exposing a great deal of error. There are a number of various solutions that are offered at a range of different levels of style abstraction in order to achieve the power and speed standards. There is support for approximate computing methodologies, which facilitate the achievement of the target specifications at the expense of a reduction in the precision of the computation [11].

This field includes applications including error-tolerant calculations, machine learning, signal processing, and the multimedia system process. The simplification of the circuits that make up arithmetic units is the primary foundation upon which approximate arithmetic units are grounded [12]. Appropriate multipliers, which have been the focus of a number of prior investigations, offer faster processing times and reduced power consumptions but worse accuracy. These multipliers have been the subject of extensive research and development.

### III. PROPOSED 4:2 COMPRESSORS

Our study presents four reconfigurable approximate 4:2 compressors with dual-quality features. Each compressor in the system has the ability to toggle between an exact and an approximate mode of operation while it is operating. Potentially used in designs of dynamically quality-customizable parallel multipliers are compressors. The suggested compressors' basic architectures consist of two parts: the approximation kinds and the supplemental types. When in precise mode, both the supplemental and approximate portions are enabled; however, when in approximate mode, only the approximate component is active.

Components not shared with the auxiliary sections are likewise indicated by the hachured box in the approximation segment. There are two distinct modes of accuracy that the proposed DQ4:2Cs can operate in: approximate and accurate. Figure 2 presents a general block schematic of the compressors that can be found in the system.

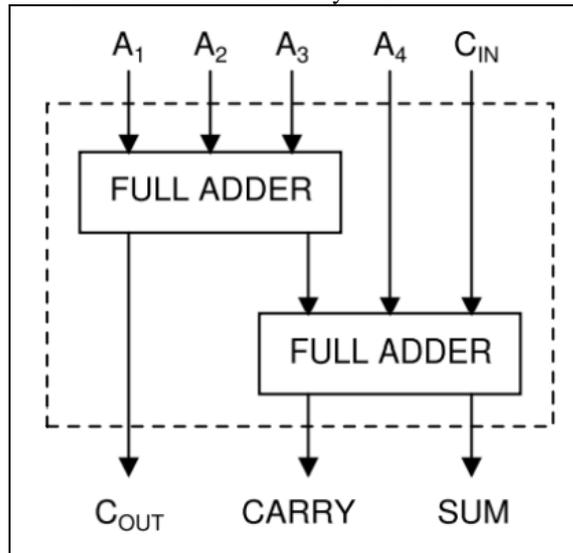


Fig.1: Exact 4:2 compressor.

### IV. RESULTS AND STUDY

#### A. RTL Schematic:

Figure 3 shows the RTL schematic, which stands for "register transfer level" and is used to compare the designed architecture to the ideal architecture that needs to be developed.

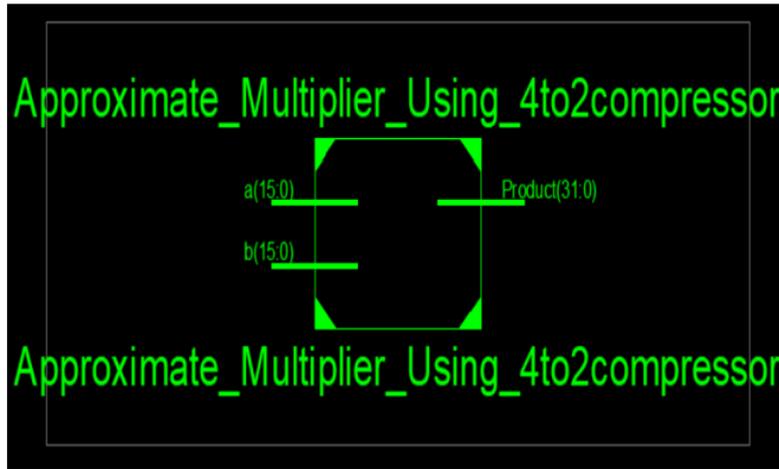


Fig. 3: RTL Schematic of the proposed design.

*B. Technology Schematic:*

In order for VLSI to estimate the architectural design, the technological schematic presents the building in LUT format, which is regarded as the parameter of area. Figure 4 shows the LUTs in an FPGA, which are essentially squares, and how the code's memory allocation is depicted in them.

*C. Simulation:*

In terms of its functionality, the simulation is the last check, whereas the schematic, depicted in figure 5, represents the verification of the blocks and connections.

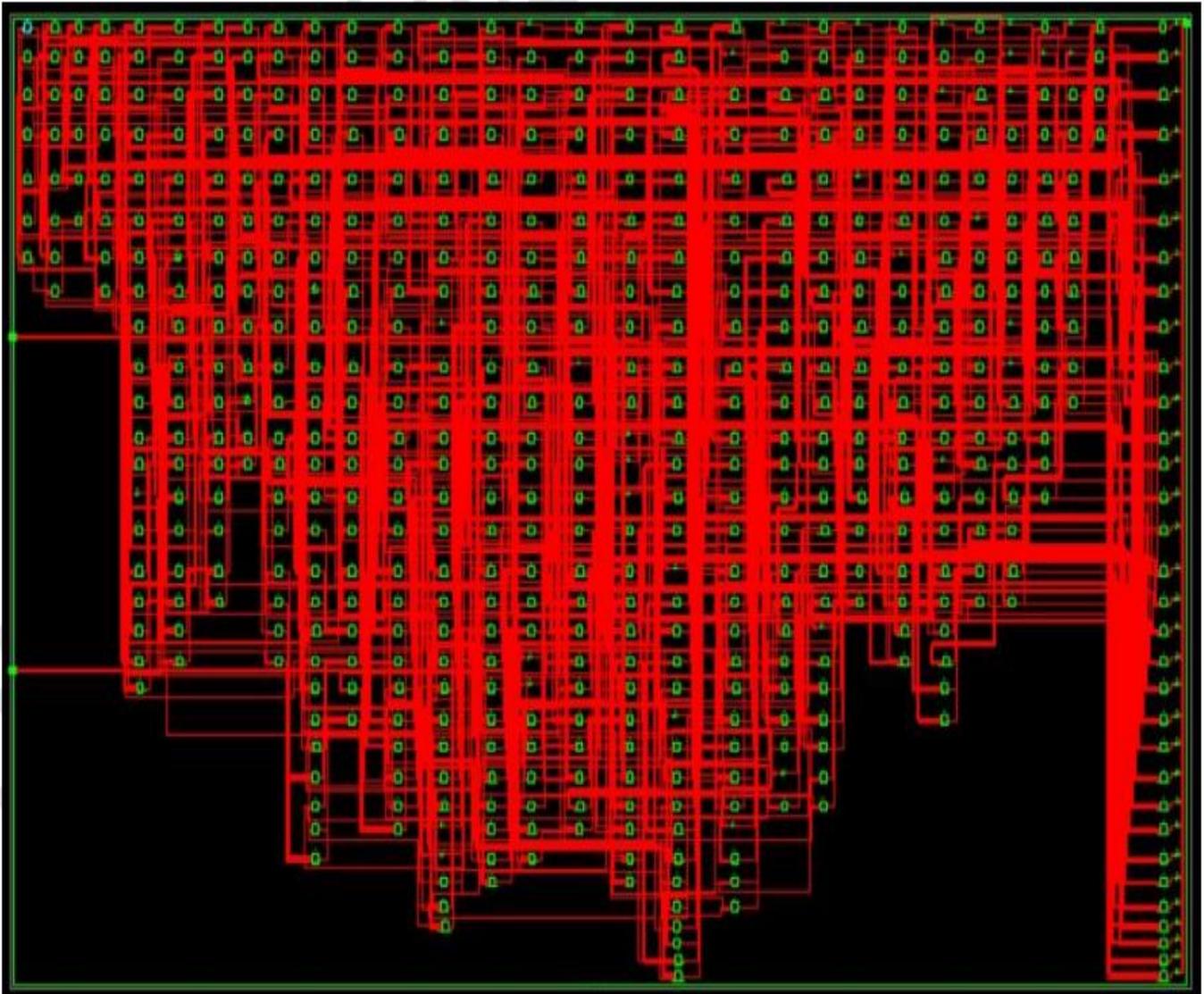


Fig 4 Technology schematic of the proposed design

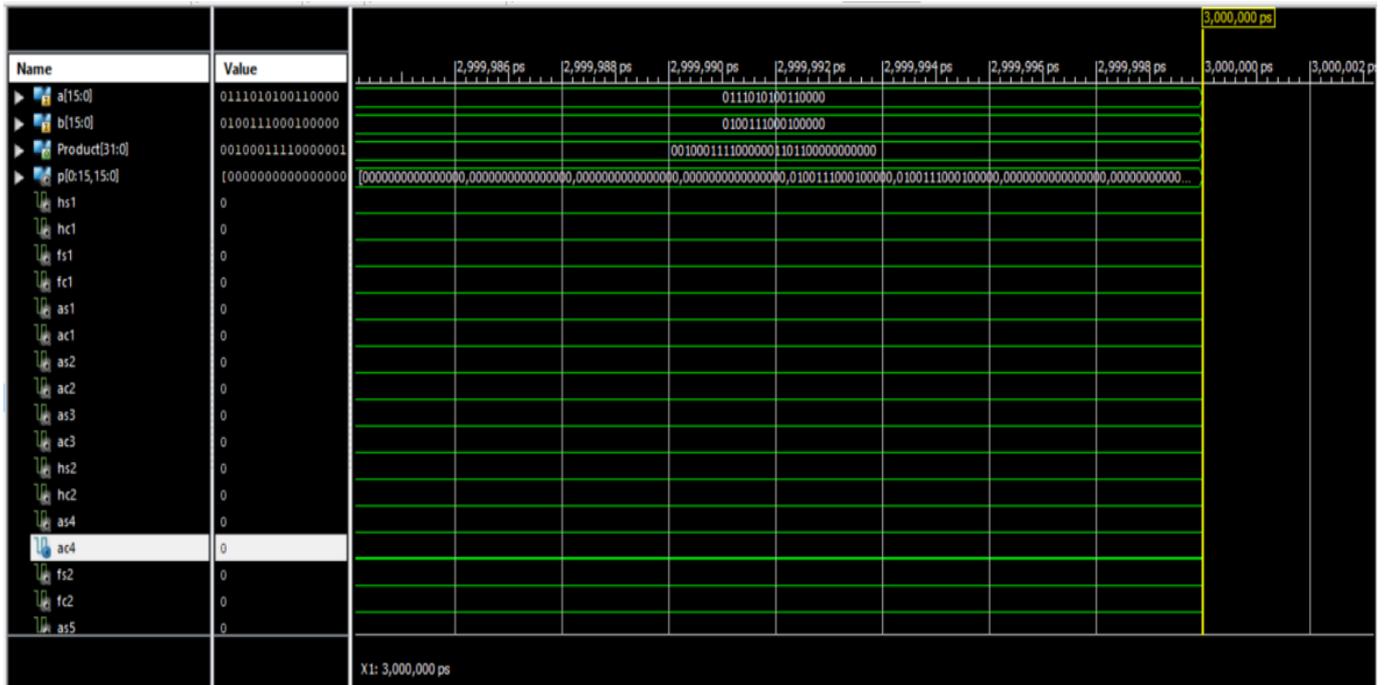


Fig 5: Simulation wave forms of proposed approximate multiplier

**IV. CONCLUSION**

This research introduces a new method of approximating 4: 2 compressor topologies, which allows for the creation of an approximation multiplier. First, we need to suggest a space-and power-efficient design for a high-speed compressor that can significantly cut down on power consumption, area, and latency as compared to current best practices. The suggested configuration is almost same. Consequently, the suggested layout lessens latency and area power consumption. Plus, despite keeping the accuracy metrics the same, the model suggested a new design for a two-stage compressor that was more space, latency, and power efficient than the old one. Designed for use in image processing tasks such as image multiplication and smoothing, the architecture showcases a 16 × 16 Dadda multiplier. Compressors vary in the amount of accuracy they achieve in the approximate mode, as well as the delays and powers that are available in the precise mode. In order to gauge how well the suggested compressors work, they are built into an 8-bit Dadda multiplier. According to our research, compared to current compressors, the predicted ones have less latency and use less power in the approximate mode when doing 8-bit multiplication.

**REFERENCES**

- [1] H. Jiang et al. A comparative evaluation of approximate multipliers
- [2] L. Cui Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things IEEE Internet of Things J. (June 2019)
- [3] W. Liu et al. Design of approximate Radix-4 Booth multipliers for error-tolerant computing IEEE Trans. Comput.(Aug. 2017)
- [4] J. Han et al. Approximate computing: an emerging paradigm for energy-efficient design
- [5] S. Venkataramani et al. Computing Approximately, and Efficiently (2015)
- [6] J. Liang et al. New metrics for the reliability of approximate and Probabilistic Adders IEEE Trans. Comput. (Sep. 2013)
- [7] Weiqiang Liu et al. Design and analysis of approximate redundant binary multipliers IEEE Trans. Comput. (Jun. 2019)
- [8] P. Kulkarni et al. Trading accuracy for power with an underdesigned multiplier architecture
- [9] K. Bhardwaj et al. Power-and area-efficient approximate Wallace tree multiplier for error-resilient systems
- [10] S. Rehman et al. Architectural-space exploration of approximate multipliers
- [11] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A reconfigurable approximate carry look-ahead adder," IEEE Trans. Circuits Syst. II, Express Briefs, doi: 10.1109/TCSII.2016.2633307.
- [12] A. Raha, H. Jayakumar, and V. Raghunathan, "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 3, pp. 846–857, May 2015.